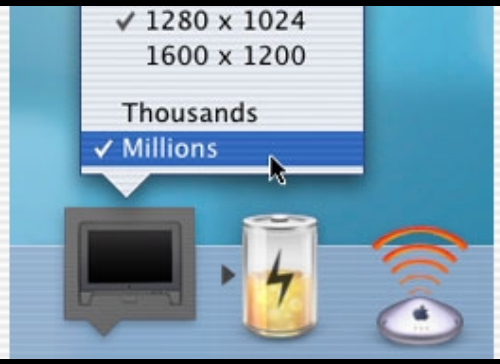
Session 106

Carbon Event Manager



Ed Voas
Carbon Event Enthusiast

What You'll Learn

- Brief Overview of Carbon Events
- How to build a Carbon Event-based app
- How to extend the Toolbox via Carbon Events

Carbon Event Basics

- Direct dispatching model
  - Receive events via Event Handlers
- Toolbox provides default behaviors
  - App overrides defaults as necessary
- WNE-based apps can use Carbon Events
  - WNE only sees events that aren't handled

Why Carbon Events?

- Greater breadth of functionality
- Unifies multiple ‘messaging’ models
- Simplifies writing applications
- Encourages better performance

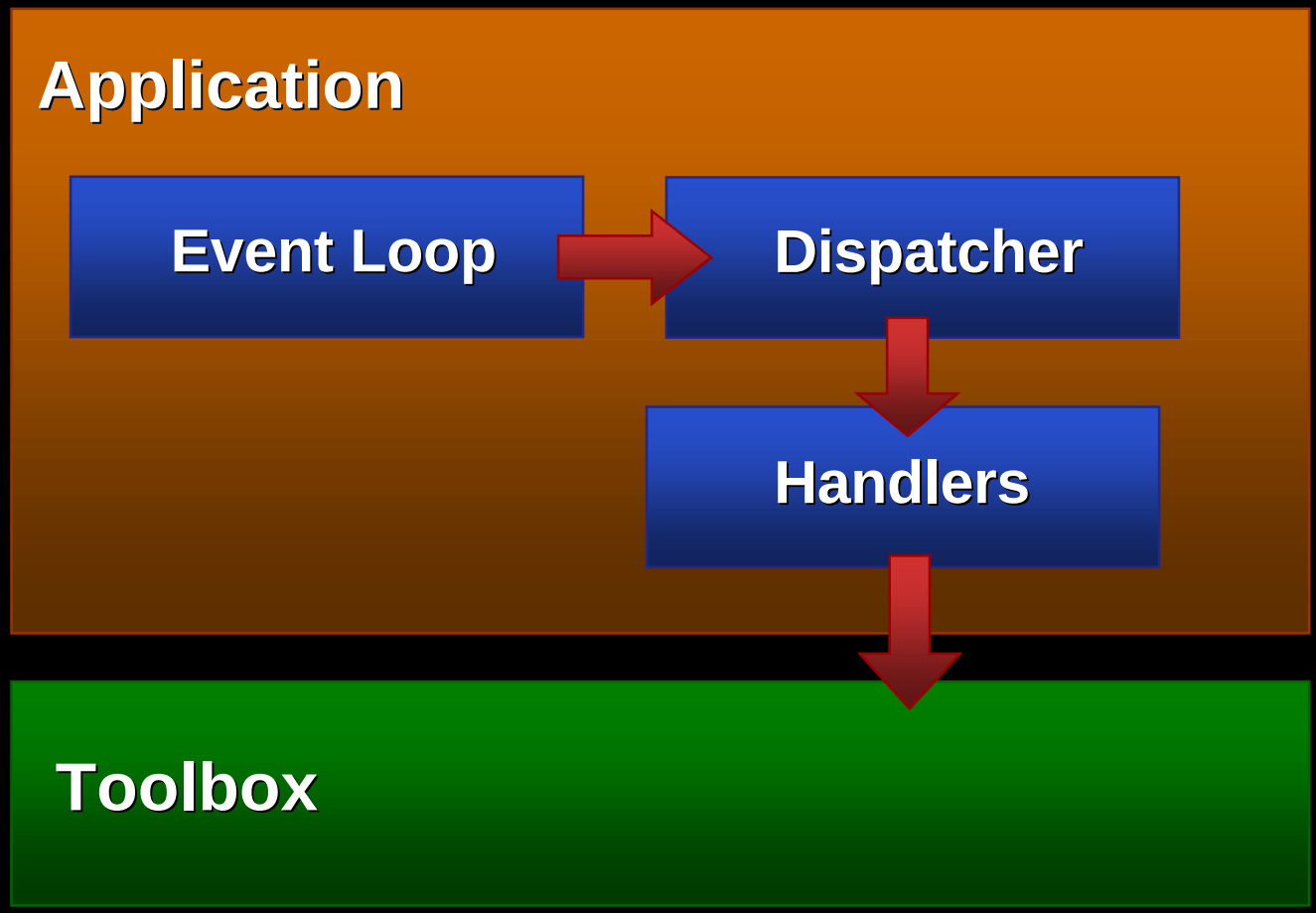
Building an Application

- Project Builder and Interface Builder
  - Shortest path to developing a Carbon Event-based app
- Practically no code to start up an app

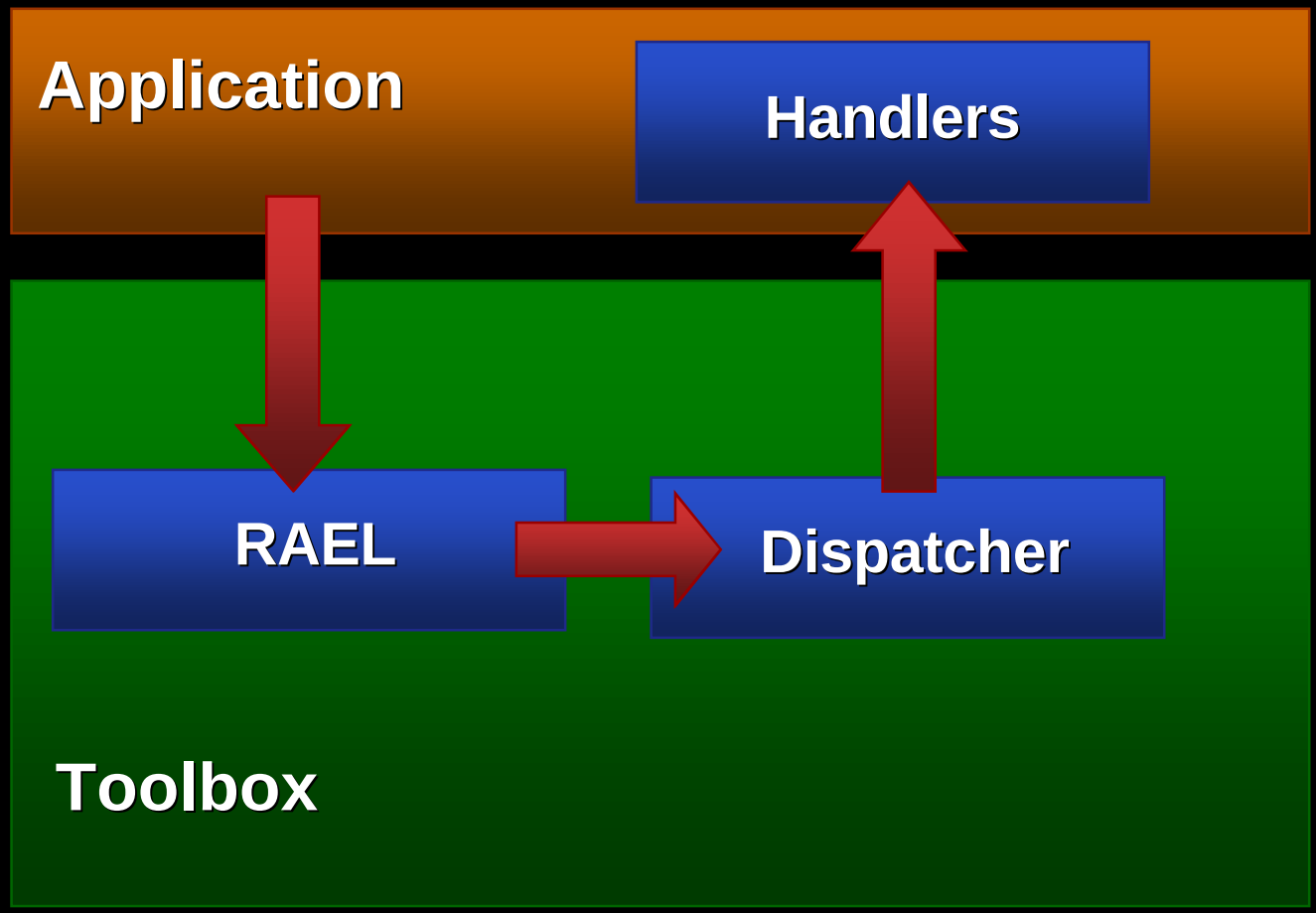
Demo

Four-Eyed FrankenGuy

1984: Ed Still in High School



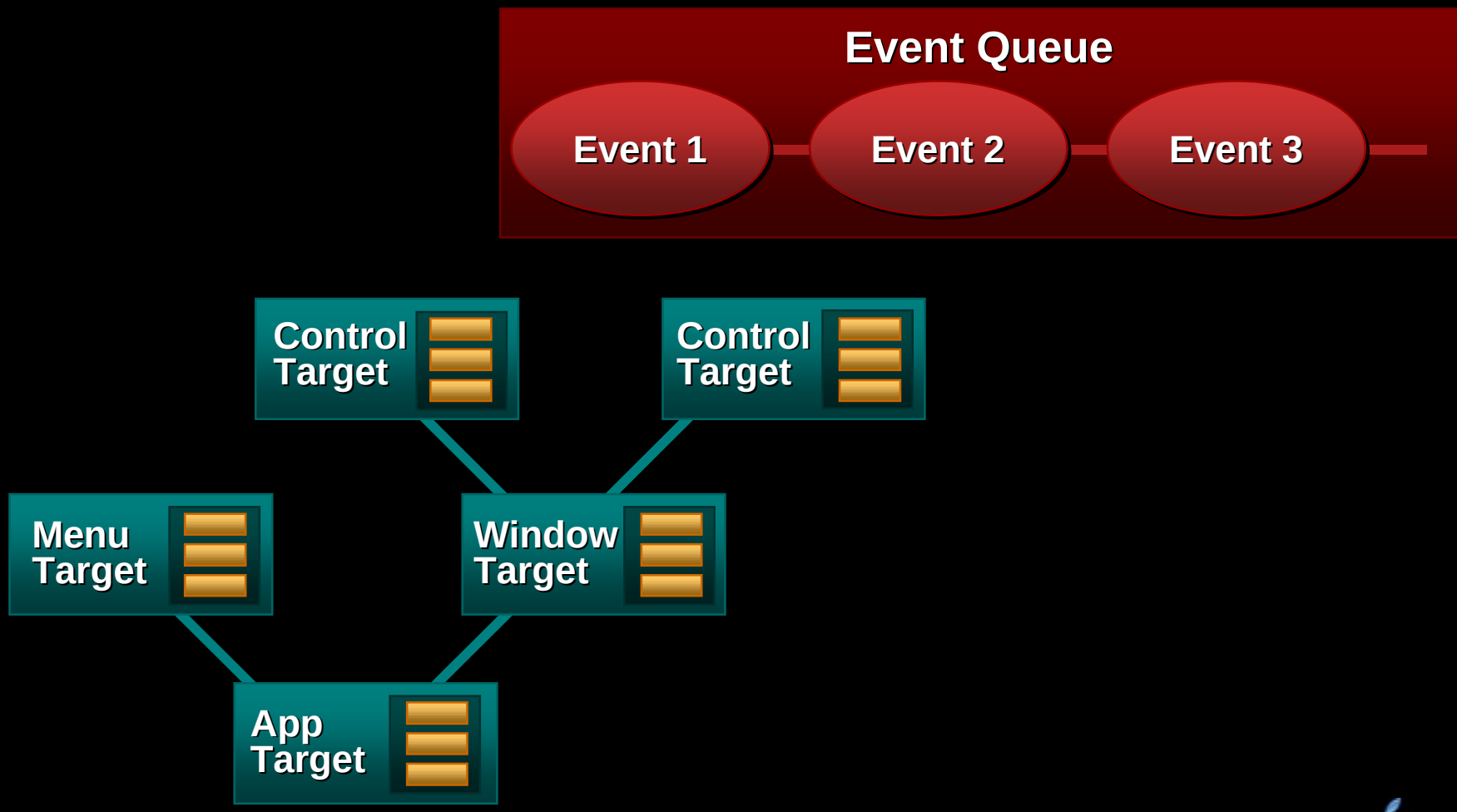
2001: Graduated 16 Years Ago



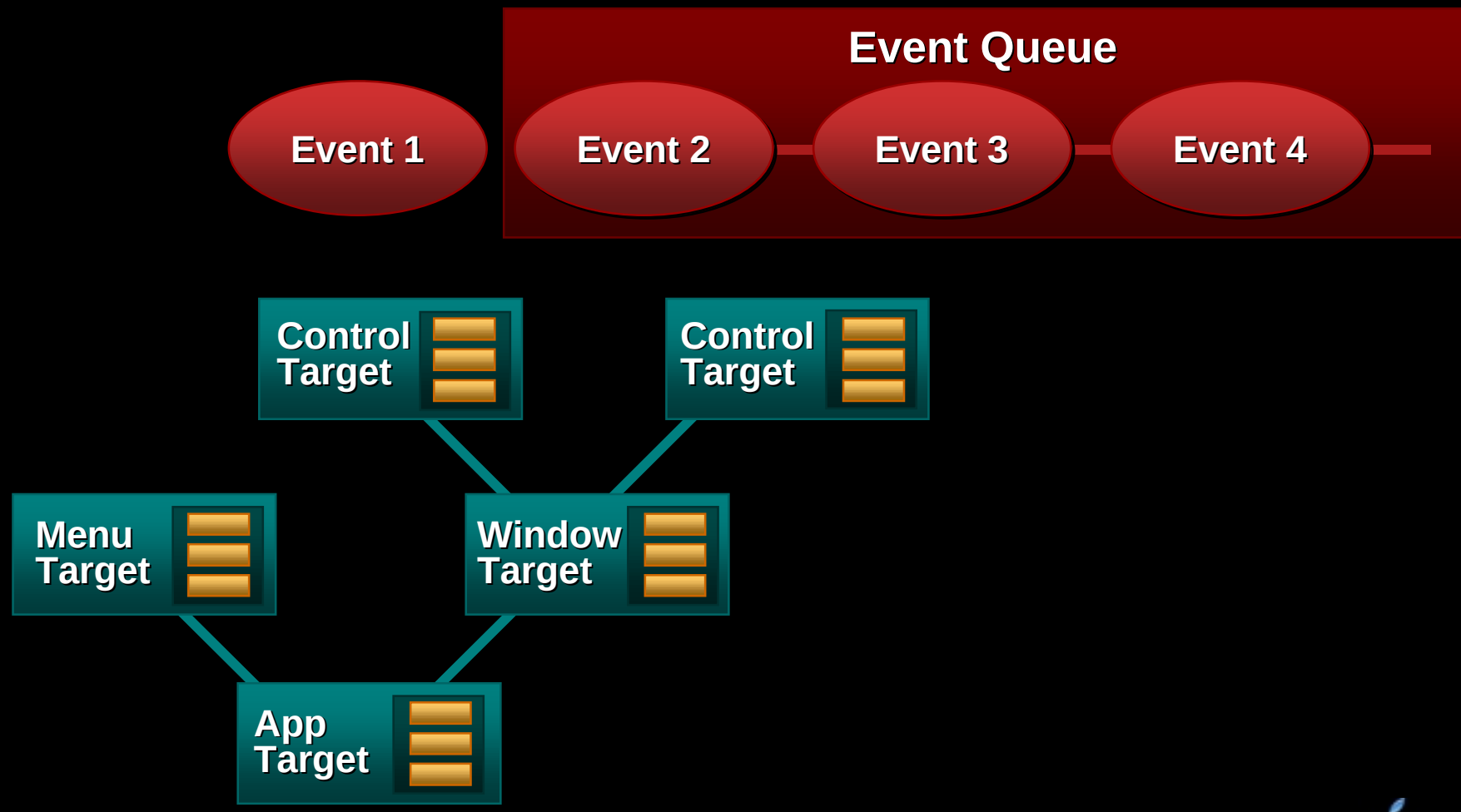
Standard Behaviors

- Windows
  - Dragging, Resizing, etc.
  - Contextual Menu Interception
- Menus
  - Menu Tracking, MenuKey
- Controls
  - Tracking, key handling

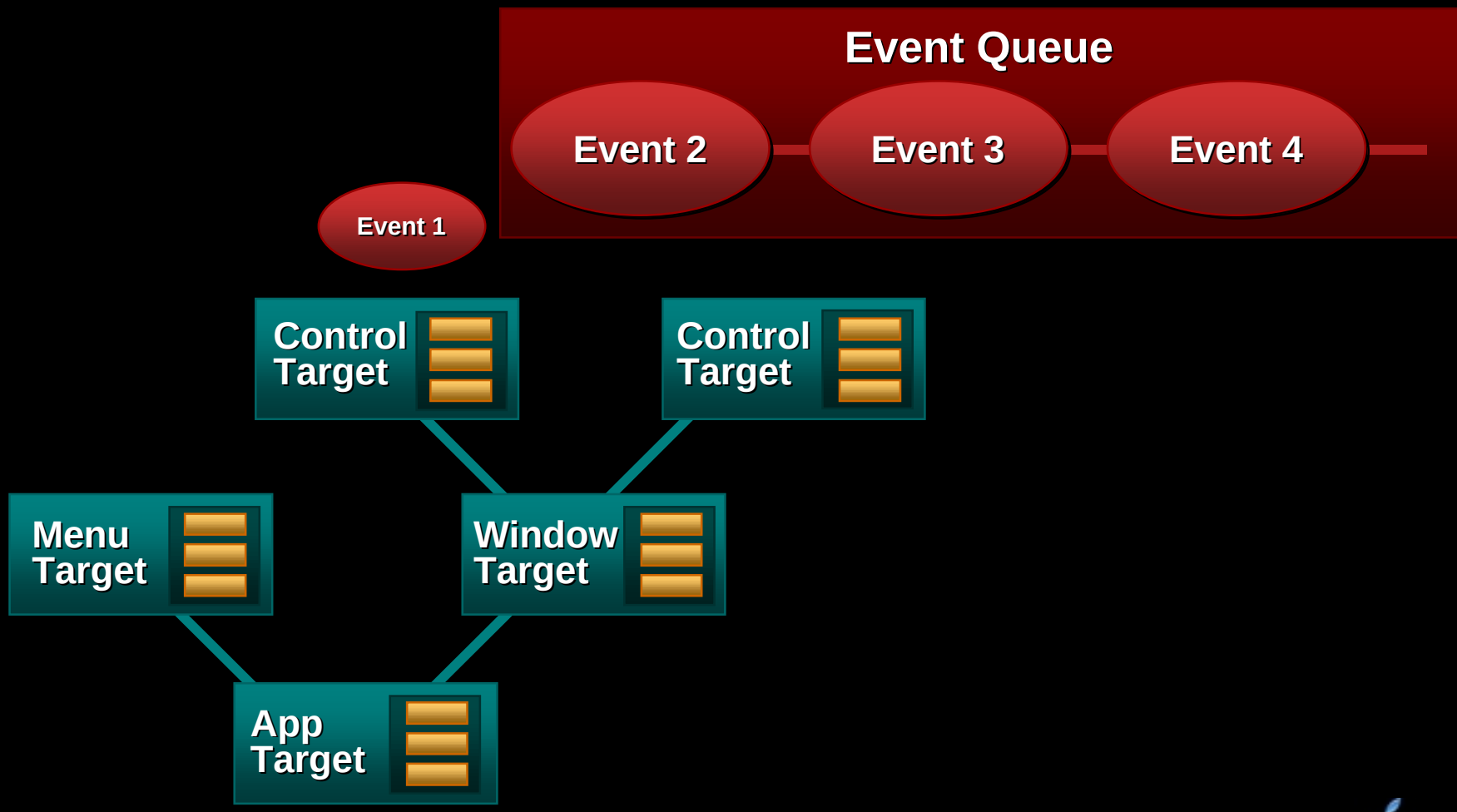
Event Flow


Event Queue
Event 1
Event 2
Event 3
Control Target
Control Target
Menu Target
Window Target
App Target

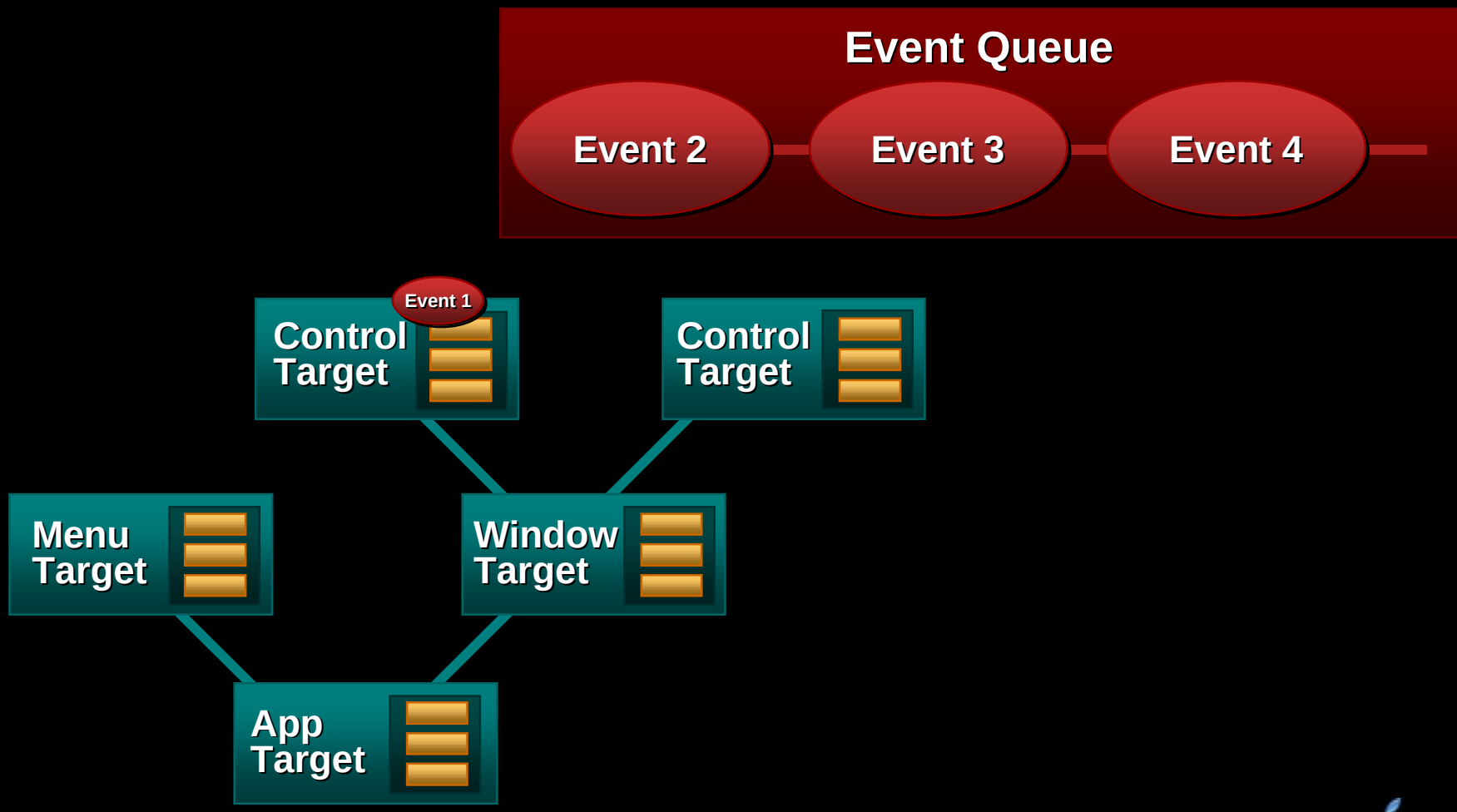
Event Flow


Event Queue
Event 1
Event 2
Event 3
Event 4
Control Target
Control Target
Menu Target
Window Target
App Target

Event Flow


Event Queue
Event 2
Event 3
Event 4
Event 1
Control Target
Control Target
Menu Target
Window Target
App Target

Event Flow


Event Queue
Event 2
Event 3
Event 4
Event 1
Control Target
Control Target
Menu Target
Window Target
App Target

Event Flow

Event 1
Control Target
My Handler
Toolbox Handler
Object Handler

Event Flow

Event 1
Control Target
My Handler
Toolbox Handler
Object Handler

Event Flow

Control
Target
Event 1
My Handler
Toolbox Handler
Object Handler

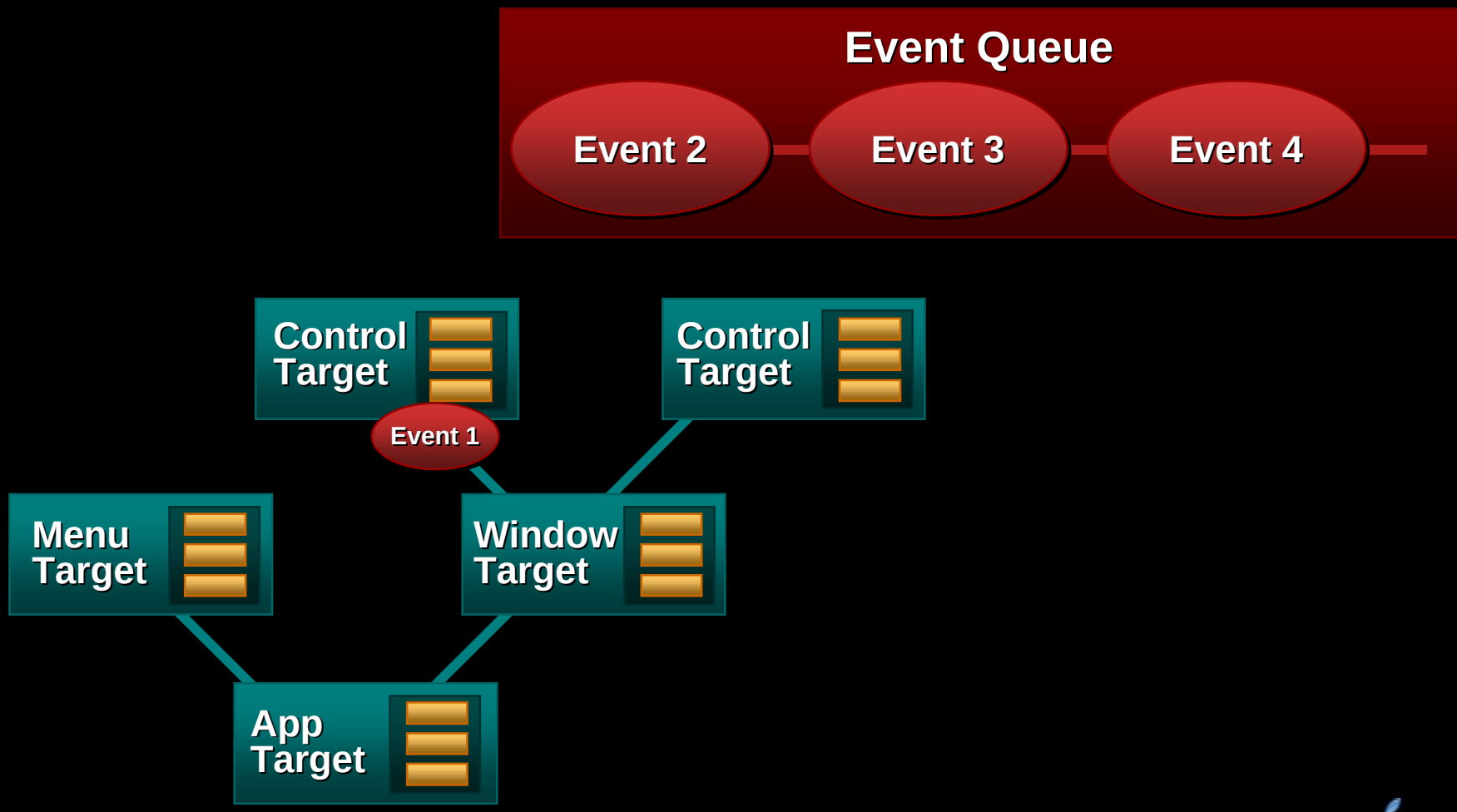
Event Flow

Control Target
My Handler
Toolbox Handler
Object Handler
Event 1

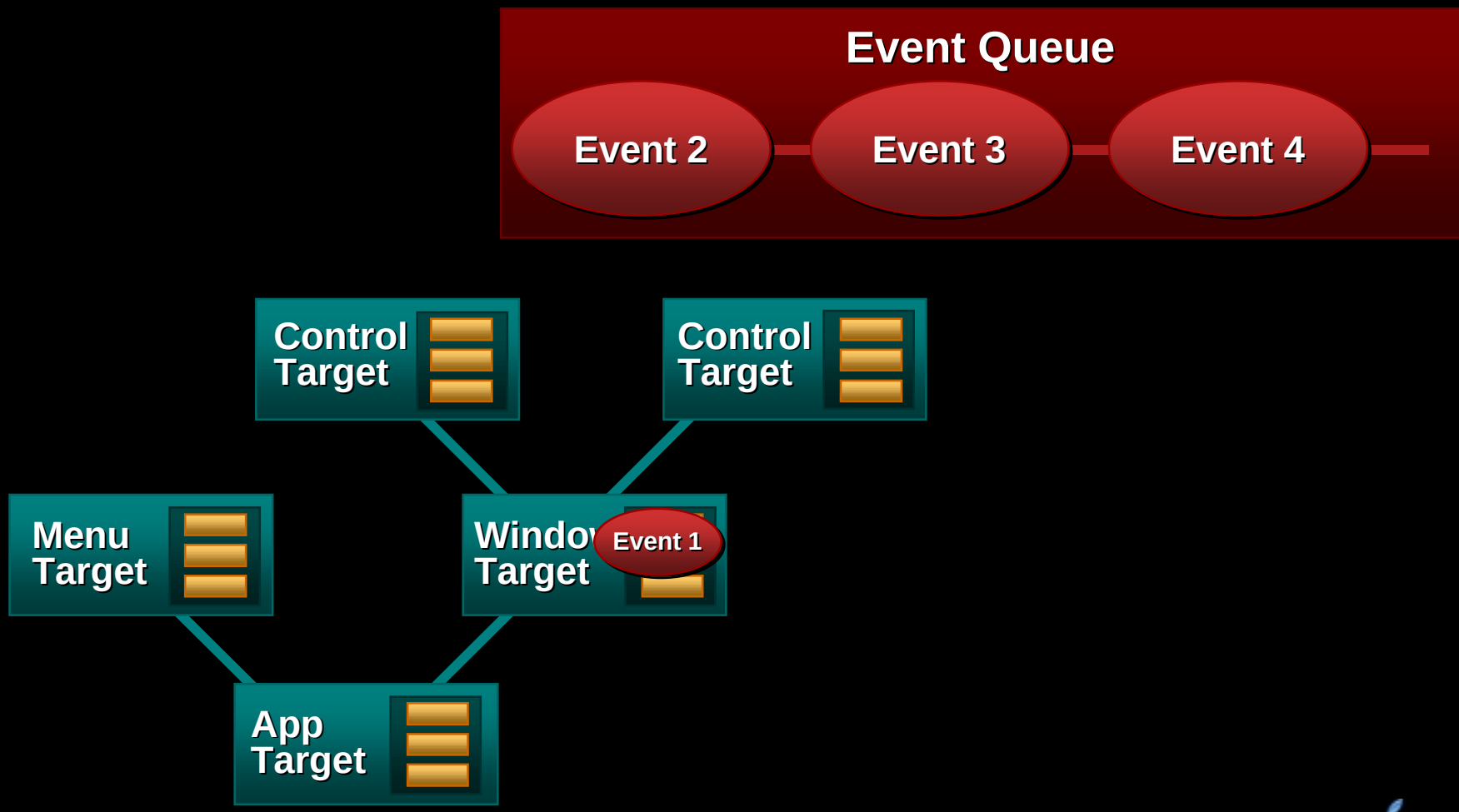
Event Flow

Control Target
My Handler
Toolbox Handler
Object Handler
Event 1

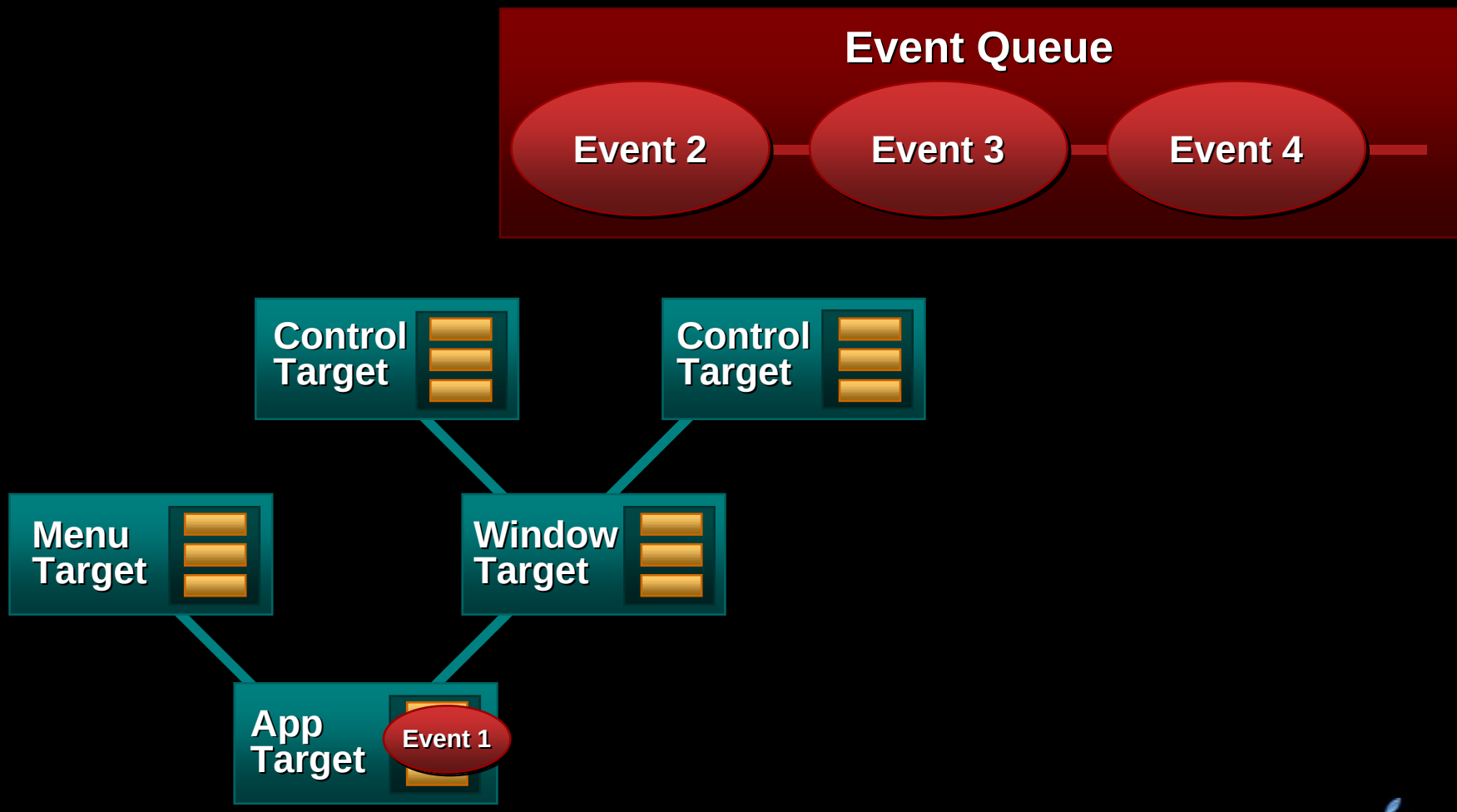
Event Flow


Event Queue
Event 2
Event 3
Event 4
Control Target
Event 1
Control Target
Menu Target
Window Target
App Target

Event Flow


Event Queue
Event 2
Event 3
Event 4
Control Target
Control Target
Menu Target
Window Target
Event 1
App Target

Event Flow


Event Queue
Event 2
Event 3
Event 4
Control Target
Control Target
Menu Target
Window Target
App Target
Event 1

EventRefs

- Transport for Carbon Events
- Identified by Class and Kind
- Named, typed parameters
  - Get/SetEventParameter
- Not necessarily a one-way street
- Apps can create their own

Event Queue

- Per-application, not shared on Mac OS X
- Thread-savvy
  - Each MPTask gets it own implicit queue
  - Can post events to communicate
- All cooperative threads share main queue
- Waiting on queue allows timers to fire

Timers

- Replacement for null events
  - Use instead of WNE timeouts
- Timers fire only when waiting on Event Queue
- Allow idle time to be given directly to an object
- Repeating or one-shot
- Timers can fire while mouse is down
- More reliable heartbeat on Mac OS X

Driving Your Application

- RunApplicationEventLoop
  - New, pure Carbon Event applications
  - Once entered, your app stays here until quit with QuitApplicationEventLoop
- WaitNextEvent
  - Legacy applications

Event Targets

- Abstract object that can receive events
- Attached to Toolbox objects
  - Windows, Controls, etc.
- Some special targets
  - User Focus
  - Toolbox Dispatcher
- Contains stack of Event Handlers

Event Handlers

- Callbacks where you receive events
- Only receive events for which you register
- Can choose to handle or not
- Result determines if events should propagate

Event Propagation

- Events proceed from handler to handler
- If not handled, they flow between targets
- Application is where all unhandled events land
- Can call through to handlers/targets below you

User Focus

- Where key events are sent
  - GetUserFocusTarget
- Definition:
  - Focused control in the currently active document window
  - If no focused control, window is focus
- Can use SetUserFocusWindow to redirect
- Also used as the target for HICommands…

HICommands

- Extension of menu commands in Mac OS 8
  - For both menus and controls now
- Different propagation paths
  - For control—control hierarchy
  - For menus—to originator, then User Focus

HICommands (Cont.)

- Facilitate menu item processing
  - kEventCommandUpdateStatus
  - kEventCommandProcess
- Inter-target messaging
  - e.g., controls send commands to their owner
- Use standard command IDs whenever possible

Window Modality

- The Easy Way:
  - Run/QuitAppModalLoopForWindow
- The Low-Level Way:
  - Begin/EndAppModalStateForWindow
  - Use if you need control over event loop

Performance

- Timers vs. Timeouts
- Tracking Loops
  - Traditional tracking loops are CPU hogs on Mac OS X
  - Use TrackMouseLocation instead
- Use notifications whenever possible

Toolbox Objects

- Replacement for procptr-based defprocs
- Register with RegisterToolboxObject
- Supported in Window and Control Managers
  - Create with CreateCustomWindow or CreateCustomControl
- Recommended method for implementing custom objects

Demo

Hockey Brokenarm
(That's Guy)

Now What?

- Start memorizing CarbonEvents.h
- Read the *Learning Carbon* book from O'Reilly
- Upcoming MacTech article
- Create a new Carbon Event application with Project Builder
- Experiment with Carbon Events—Have fun!

Roadmap

112: Carbon Windows & Menus Learn about sheets, window groups, etc.	Hall 2 **Wed., 2:00pm**
119: Carbon Controls & Appearance I Adopting Aqua	Hall 2 **Thurs., 9:00am**
120: Carbon Controls & Appearance II Don't let the title fool you: help tags, too!	Hall 2 **Thurs., 10:30am**
121: Carbon Performance Tuning Learn how to perform well on Mac OS X	Hall 2 **Thurs., 2:00pm**

WWDC

Worldwide Developers Conference 2001